Indexing an image database by shape content usin curvature scale space

Mokhtarian, F. Abbasi, S. Kittler, J.
Dept. of Electron. & Electr. Eng., Surrey Univ., Guildford

This paper appears in: **Intelligent Image Databases, IEE Colloquium on**

Meeting Date: 05/22/1996
Publication Date: 22 May 1996
Location: London UK
On page(s): 4/1 - 4/6
Reference Cited: 7
Inspec Accession Number: 5297826

Abstract:
We introduce a very fast and reliable algorithm for shape similarity retrieval i image databases which is robust with respect to noise, scale and orientation c the objects. The **curvature scale space** image is used to represent the shap **boundary contours**. Since the algorithm uses the global information of bour objects, it is sensitive to major occlusion, but some minor occlusion can be de the algorithm. We tested our method on a database of 450 images of marine with a vast variety of shapes, with very good results which we present in this

Index Terms:
curvature scale space **curvature scale space** image image database indexing in retrieval marine animals object **boundary contours** shape content shape similarity visual databases **curvature scale space** **curvature scale space** image image datal indexing information retrieval marine animals object **boundary contours** shape coi similarity retrieval visual databases

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

INDEXING AN IMAGE DATABASE BY SHAPE CONTENT USING CURVATURE SCALE SPACE

Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler [1]

Abstract

We introduce a very fast and reliable algorithm for shape similarity retrieval in large image databases which is robust with respect to noise, scale and orientation changes of the objects. The Curvature Scale Space image is used to represent the shapes of object boundary contours. Since the algorithm uses the global information of boundaries of objects, it is sensitive to major occlusion, but some minor occlusion can be detected by the algorithm.

We tested our method on a database of 450 images of marine animals with a vast variety of shapes, with very good results which we present in this paper.

1 Introduction

Most proposed content based database systems aim to retrieve a small set of candidate images which include the desired image. The successful retrieval of the best candidate then relies on the final user judgement. In [1], the authors have used Polygonal approximation, while a set of features like boundary/perimeter, elongation (major axis/ minor axis), number of holes, etc, have been used in [2] for shape similarity retrieval. The authors in [3] have used a combination of heuristic shape features such as area, circularity, eccentricity, major axis orientation and a set of algebraic moment invariants. They have also used other features such as color, texture, and even sketch features.

We use a modified version of Curvature Scale Space image matching [6] for comparing shapes of objects in an image database. Our prototype database includes more than 450 colored images of marine animals, with every image containing one animal. The preprocessing step (consisting of gray-level morphology, thresholding and binary morphology) extracts the boundaries of objects. Other techniques such as active contours [4] can also be incorporated at this stage if necessary.

We compute the CSS image of every boundary and then find the maxima of CSS contours which are used as a shape descriptor to compare objects. The coordinates of these points together with the name of the original image constitute a record which represents the object.

To retrieve similar images from the database, the user can either input an image and ask the system to find all images similar to it or sketch a boundary of his/her desired object using a painting package such as *xpaint*. The system computes the CSS image of the input and finds its maxima, and after comparison, assigns a matching value to every image candidate in the database which is similar to the input and shows the first n matched images with best values where n is determined by the user.

During query processing we use our fast algorithm to compare the input and the candidate representations and assign a matching value to every candidate.

[1] **The authors are with the Department of Electronic and Electrical Engineering at the University of Surrey, Guildford**

The Curvature Scale Space Representation

The curvature of a curve is defined as the derivative of the tangent angle to the curve. Consider a parametric vector equation for a curve:

$$\vec{r}(u) = (x(u), y(u))$$

where u is an arbitrary parameter. The formula for computing the curvature function can be expressed as[5]:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}} \quad . \tag{1}$$

If Γ is a closed planar curve, u can be the normalized arc length parameter which means:

$$\Gamma = \{ (x(u), y(u)) \mid u \in [0,1] \}$$

and the denominator in equation (1) will then be equal to one and we obtain:

$$\kappa(u) = \dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u) \quad .$$

In the rest of this paper we will assume that the curves are closed and planar and are expressed in terms of the normalized arc length.

Curvature zero crossings of a curve are points with zero curvature. On smooth curves, wherever the sign of curvature changes, there is a point with zero curvature.
There are several approaches in calculating the curvature of a digital curve [7]. We use the idea of *curve evolution* which studies shape properties while deforming in time. A certain kind of evolution can be achieved by Gaussian smoothing to compute curvature at varying levels of detail. If $g(u,\sigma)$ is a 1-D Gaussian kernel of width σ, then $X(u,\sigma)$ and $Y(u,\sigma)$ represent the components of *evolved* curve,

$$X(u,\sigma) = x(u) * g(u,\sigma) \qquad Y(u,\sigma) = y(u) * g(u,\sigma)$$

where $*$ is the convolution operator. According to the properties of convolution, the derivatives of every component can be calculated easily :

$$X_u(u,\sigma) = x(u) * g_u(u,\sigma) \qquad X_{uu}(u,\sigma) = x(u) * g_{uu}(u,\sigma)$$

and we will have a similar formula for $Y_u(u,\sigma)$ and $Y_{uu}(u,\sigma)$. Since the exact forms of $g_u(u,\sigma)$ and $g_{uu}(u,\sigma)$ are known, the curvature of an evolved digital curve can be computed easily.

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}} \tag{2}$$

As σ increases, the shape of Γ_σ changes. This process of generating ordered sequences of curves is referred to as the evolution of Γ.

2 Curvature Scale Space Matching

We assume that the user enters his query by sketching a boundary of his desired image or by pointing to an image and wants the system to retrieve all images like that one from the database. In each case, we do the same preprocessing to find the maxima of CSS contours of input image and compare them with the same descriptors of the database images. For convenience, from now on, we call the input as *image* and the images in the database as *models*. In this section we explain the algorithm of matching which is rather different from what is proposed in [6].

Algorithm

After extracting maxima of every model, we normalize their location so that the horizontal coordinate u varies in the range $[0, 1]$. This will ensure that the comparison is meaningful even if the number of samples in image and model are different. The maxima of every model are sorted according to their σ-coordinate (filter width) during the process of maxima extraction.

The matching algorithm which compares the two sets of maxima, one from the image and the other from the model is as follows.

1. Create a node consisting of the largest scale maximum of the image and the largest scale maximum of the model. If there are more than one maxima in the model which have a σ-coordinate close (within 80 percent) to the largest scale maximum of the image, create extra nodes consisting of the largest scale maximum of the image and that respective additional maxima of the model.

   Initialize the *cost* of each node to the absolute difference of σ-coordinates of the image and the model.

   Compute a CSS shift parameter α for each node :

   $$\alpha = U_m - U_i$$

   where U is the horizontal coordinate of a maximum, and i and m refer to image and model respectively. If the model and the image are the same and we shift the CSS of image horizontally by α, the two CSS should cover each other.

2. Create two lists for each node obtained in step 1. The first list will contain the model curve maxima and the second list will contain the model curve maxima matched within that node at any point of the matching procedure. Initialize the first and second list of each node by the corresponding maxima determined in step 1.

3. Expand each node created in step 1 using the procedure described in step 4.

4. To expand a node, select the largest scale image curve CSS maximum (which is not in the first list) and apply that node's shift parameter computed in step 1 to map that maximum to the model CSS image. Locate the nearest model curve CSS maximum (which is not in the second list). If the two maxima are in a reasonable horizontal distance (0.2 of the maximum possible distance), define the cost of the match as the straight line distance between the two maxima. Otherwise, define the height of the image curve CSS maximum as the cost of the match. If there are no more image curve CSS maxima left, define the cost of match as the height of the highest model curve CSS maximum *not* in the node's second list. Likewise, if there are no more model curve CSS maxima left, define the cost of match as the height of the selected image curve maximum. Add the match cost to the node cost. Update the two lists associated with the node.

5. Select the lowest cost node. If there are no more model or image curve CSS maxima that remain unmatched within that node, then return that node as the lowest cost node. Otherwise, go to step 4 and expand the lowest cost node.

6. Reverse the place of the image and the model and repeat steps 1 to 5 to find the lowest cost node in this case.

7. Consider the lowest node as the final matching cost between the image and the model.

3 Results and discussion

We tested the proposed method on a database of 450 images of marine animals. Each image consisted of just one object on a uniform background. The system software was developed using the C language under Unix operating system. The response rate of the system was less than one second for every user query.

After extracting the contour of the object in each image, it was sampled at 200 equal distance points. The normalized coordinates of these points were then used to find the CSS image of the contour . This procedure was followed by the extraction the maxima of every CSS image.The coordinates of these points were stored in a record together with the number of rows and columns of the CSS image and the name of the original image in a file which was read at the beginning of every query processing. We discuss the experimental results by representing the response of the system to some queries. In the first one the user has shown an image which really existed in the database and asked for similar images. The input image is in figure 2a and the output of the system is in figure 3a . The first answer of the system is identical with the input image, with a zero match value. For the remaining images, the match value varies between 0.29 to 0.38 . When the match value is more than 0.45, the similarity between input and output images is poor. Note that the fourth and fifth output images are different just in scale.

In the second query, user has used mouse to paint an outline of his desired image. Figure 2b is the input which is apparently noisy, and figure 3b consists of retrieved images. As this example shows, the system is robust with respect to noise, because the location of maxima on CSS is identified after a process of smoothing and this process for the existing system begins with a Gaussian filter with $\sigma = 1$. Although this property is beneficial in most cases like this example, sometimes, eg when there are some ripples on the contour of the original image, it may remove some useful information and even some maxima of CSS image. In general, there must be a limit on the initial value of σ so that the noise is removed and the useful information is not lost.

Other examples are presented in figures 3c and 3d, where the inputs and the first outputs of the system are identical and other outputs are similar to the inputs. These examples also show the variety of shapes of objects in our database.

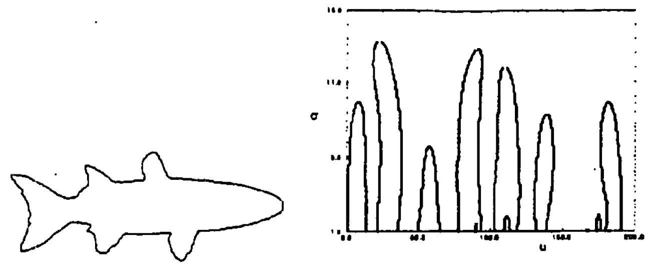
4 Conclusions

This paper described a method to retrieve similar images from large image databases using their shape properties. A database of 450 images of marine animals was selected to test the method. The boundary of every object was extracted and the curvature scale space image of this boundary was computed. Then the maxima of this image were extracted to be considered as the shape descriptors. A fast and reliable algorithm for comparing these descriptors together with the results of several queries were presented.

Acknowledgements

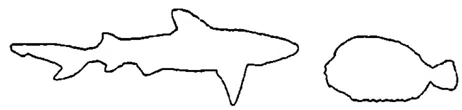
Sadegh Abbasi is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his research studies.

The morphology and hysteresis thresholding programs used for preprocessing of images were written by Majid Mirmehdi and Radek Marik respectively, to whom the authors are grateful.



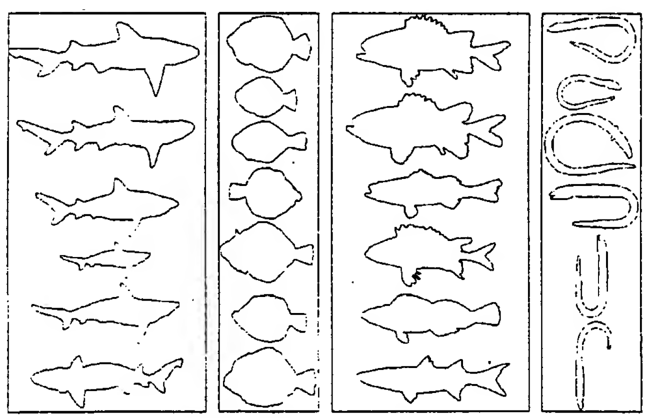
(a) (b)

Figure 1: a) a boundary and, b) its CSS image



(a) (b)

Figure 2: Two types of query: a) user points to an image, b) user paints a boundary



(a) (b) (c) (d)

Figure 3: query results

References

[1] E. Gary and R. Mehrorta, " Shape similarity-based retrieval in image databases " in SPIE, Vol. 1662, 1992.

[2] P. Eggleston, " Content based feature indexing and retrieval for image databases " in SPIE, Vol. 1819, 1992.

[3] W. Niblack *et al* , " The QBIC project; querying images by content using color , texture and shape " in SPIE, VOL. 1908, 1993.

[4] A. A. Amini, Terry E. Weymouth and Ramesh C. Jain, " Using dynamic programming for solving variational problems in vision " in IEEE·Trans. on Pattern Anal. Mach. Intell. , VOL. 12, No. 9, September 1990 pp855-866.

[5] Mokhtarian, F. and Mackworth, A. K. " A theory of multiscale, curvature-based shape representation for planar curves " in IEEE Trans. on Pattern Anal. Mach. Intell. , VOL. PAMI-14, No. 8, August 1992 pp789-805.

[6] Mokhtarian, F. " Silhouette-based isolated object recognition through curvature scale space " in IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1995, Vol.17, No.5, pp.539-544

[7] Tsai,D. and Chen M. " Curve fitting approach for tangent angle and curvature measurement " in Pattern Recognition, Vol 27, No 5, pp 699-711, 1994.

Your search matched **15** of **995179** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevanc** **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

curvature scale space <paragraph> (contour or outline [Search]

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Affine-similar shape retrieval: application to multiview 3-D object recognition**
Abbasi, S.; Mokhtarian, F.;
Image Processing, IEEE Transactions on , Volume: 10 , Issue: 1 , Jan. 2001
Pages:131 - 139

[Abstract] [PDF Full-Text (192 KB)] **IEEE JNL**

---

2 **Robust image corner detection through curvature scale space**
Mokhtarian, F.; Suomela, R.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 20 , Issue: 12 , Dec. 1998
Pages:1376 - 1381

[Abstract] [PDF Full-Text (3908 KB)] **IEEE JNL**

---

3 **Silhouette-based isolated object recognition through curvature scal space**
Mokhtarian, F.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 17 , Issue: 5 , May 1995
Pages:539 - 544

[Abstract] [PDF Full-Text (580 KB)] **IEEE JNL**

---

4 **Shape based leaf image retrieval**
Wang, Z.; Chi, Z.; Feng, D.;
Vision, Image and Signal Processing, IEE Proceedings- , Volume: 150 , Issue: 1 , Feb. 2003
Pages:34 - 43

[Abstract] [PDF Full-Text (623 KB)] **IEE JNL**

---

5 **Robust corner tracking for multimedia applications**
Mohanna, F.; Mokhtarian, F.;
Image Processing. 2002. Proceedings. 2002 International Conference on , Vo
3 , 24-28 June 2002
Pages:945 - 948 vol.3

[Abstract] [PDF Full-Text (423 KB)] **IEEE CNF**

---

6 **Hand pose recognition using curvature scale space**
Chin-Chen Chang; I-Yen Chen; Yea-Shuan Huang;
Pattern Recognition, 2002. Proceedings. 16th International Conference on , Volume: 2 , 11-15 Aug. 2002
Pages:386 - 389 vol.2

[Abstract] [PDF Full-Text (315 KB)] **IEEE CNF**

---

7 **Fuzzy integral for leaf image retrieval**
Zhiyong Wang; Zheru Chi; Dagan Feng;
Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE Internatic
Conference on , Volume: 1 , 12-17 May 2002
Pages:372 - 377

[Abstract] [PDF Full-Text (433 KB)] **IEEE CNF**

---

8 **Multi-view free-form 3-D object retrieval with incomplete data**
Mokhtarian, F.; Abbasi, S.;
Multimedia Signal Processing, 2001 IEEE Fourth Workshop on , 3-5 Oct. 2001
Pages:287 - 292

[Abstract] [PDF Full-Text (426 KB)] **IEEE CNF**

---

9 **Automatic view selection in multi-view object recognition**
Abbasi, S.; Mokhtarian, F.;
Pattern Recognition, 2000. Proceedings. 15th International Conference on , Volume: 1 , 3-7 Sept. 2000
Pages:13 - 16 vol.1

[Abstract] [PDF Full-Text (336 KB)] **IEEE CNF**

---

10 **Diatom contour analysis using morphological curvature scale spac**
Wilkinson, M.H.F.; Roerdink, J.B.T.M.; Droop, S.; Bayer, M.;
Pattern Recognition, 2000. Proceedings. 15th International Conference on , Volume: 3 , 3-7 Sept. 2000
Pages:652 - 655 vol.3

[Abstract] [PDF Full-Text (372 KB)] **IEEE CNF**

---

11 **Multi-scale spline-based contour data compression and reconstruc through curvature scale space**
Yoke Khim Ung; Mokhtarian, F.;

Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 20
IEEE International Conference on , Volume: 6 , 5-9 June 2000
Pages:2123 - 2126 vol.4

[Abstract] [PDF Full-Text (248 KB)] **IEEE CNF**

---

12 **Shape similarity retrieval under affine transform: application to mu view object representation and recognition**
Abbasi, S.; Mokhtarian, F.;
Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on , Volume: 1 , 20-27 Sept. 1999
Pages:450 - 455 vol.1

[Abstract] [PDF Full-Text (3520 KB)] **IEEE CNF**

---

13 **Curvature scale space for robust image corner detection**
Mokhtarian, F.; Suomela, R.;
Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on , Volume: 2 , 16-20 Aug. 1998
Pages:1819 - 1821 vol.2

[Abstract] [PDF Full-Text (168 KB)] **IEEE CNF**

---

14 **Curvature scale space for image point feature detection**
Mokhtarian, F.; Suomela, R.;
Image Processing And Its Applications, 1999. Seventh International Conferen (Conf. Publ. No. 465) , Volume: 1 , 13-15 July 1999
Pages:206 - 210 vol.1

[Abstract] [PDF Full-Text (468 KB)] **IEE CNF**

---

15 **Indexing an image database by shape content using curvature scal space**
Mokhtarian, F.; Abbasi, S.; Kittler, J.;
Intelligent Image Databases, IEE Colloquium on , 22 May 1996
Pages:4/1 - 4/6

[Abstract] [PDF Full-Text (232 KB)] **IEE CNF**
